MT. SAN ANTONIO COLLEGE, WALNUT

Math/Computer Science Department
Fall 2019
Syllabus for CSCI 220 -- 22183
Data Structures I
Prof. Tuan A Vo
MW 1:45 pm - 4:10 pm, Room 61-1418

Course Description

Abstract data types and running time analysis tools. Linear data structures including sets, stacks, queues, and linked lists. Trees, binary search trees, heaps, and priority queues. Many procedures are discussed using an algorithmic language and selected problems are programmed in a higher level language. Prerequisite: CSCI 140 or CSCI 145 or equivalent programming experience.

Student Learning Outcomes (SLO)

1. Students will be able to analyze problems and select the appropriate data structure.
2. Students will be able to estimate running time given an algorithm.
3. Students will be able to implement and use linear data structures including sets, stacks, queues, and lists.
4. Students will be able to implement and use trees including binary tree, binary search trees, and heaps.

Course Measurable Objectives (CMO)

1. Analyze problems and select the appropriate data structure.
2. Design the most efficient data structure for solving a problem.
3. Implement the data structure through effective C++/Java code.
4. Utilize effective search, insertion and deletion algorithms.
5. Demonstrate effective debugging techniques.
6. Write and organize documentation for data structures.
7. Estimate running time for the algorithm studied in class or new algorithms.

Office Hours and Addresses

Office Location:	61-1654
Office Hours:	MW 1:00 pm – 1:45 pm TTh 7:15 am – 8:00 am, Th 1:45 pm – 2:45 pm
Office Phone:	(909) 274-4519
Email:	tvo@mtsac.edu
Web Page:	http://zeus.mtsac.edu/~tvo
Course Link:	https://myportal.mtsac.edu (via Canvas)

Textbooks and Materials

- M. Goodrich, R. Tamassia, and D. Mount, *Data Structures and Algorithms in C++*, Second Edition, Wiley, 2011 (can also use the 6th edition of the Java book as well).
- Access to a computer with a C++/Java compiler (available in Math/CSCI computer lab).
- Some USB storages to store and submit your work.

Grading

Grading will be based on homework assignments, pop quizzes/in-class exercises, exams, and projects. The final grade is generally based on a straight scale:

A >= 90%, 80% <= B < 90%, 70% <= C < 80%, 60% <= D < 70%, F < 60%.

Percent	Items	Lecture Points	Lab Points
9%	Homework -- 3	90	
12%	Labs -- 12		120
18%	Projects -- 4		180
4%	Pop Quizzes/Class Exercises -- 8	40	
30%	Exams -- 2	300	
6%	Lab Final -- comprehensive		60
21%	Written Final -- comprehensive	210	

All assignments must be turned in at the beginning of the class session on the due date. Late assignments will only be accepted up to the beginning of the next class session with a 10% deduction with no exceptions. All exams will cover the homework assignments, textbook reading assignments, lab, projects, and class activities. The exams will usually consist of true/false, multiple choice, problem solving, and short essay questions. Tests can only be made up with instructor's prior approval or special circumstance that can be substantiated.

Some potential extra credit points can be earned from projects, class activities, and excellent work. A pop quiz or class exercise can be given any time (must be there when it is given out and no make-up pop quizzes) so make sure to come to class on time and be there for the duration of the class.

Lab Activities

The goal of the lab is to enhance teaching/learning and to develop a deeper understanding of concepts taught in class. We will use lab time to lecture on topics, demonstrate new concepts via programming examples, and you would have time to try out new concepts under the supervision of your professor. Lab activities would include going over new concepts, solving problems, modifying existing programs, testing programs, debugging programs, and conducting experiments. Important concepts will be demonstrated so you can complete your labs and programming projects. Besides coding analysis, design, testing and debugging it will be important that you learn individually or in collaborative group work. You might be asked to present/share your solutions with other students so giving you a chance to learn from others.

Students need to prepare for lab by reading the material assigned and covered in lecture. Each lab will start with a preview of the concepts on which the lab is based in one of the following forms: instructor lecture or demonstration, class questions or discussion, using class worksheets: T/F, fill-

in blanks, or short answers. Class lab handouts will guide you to use the lab resources, lab textbook or interactive visualization software. Handouts on specific additional topics, code skeletons or demonstrations will be provided.

As you work on code, it is recommended that you talk to each other (unless otherwise instructed) or ask your instructor for individual help. If a topic will be of interest for the whole class we will take it up for class discussion and your instructor will lecture or demonstrate to clarify that topic. When you create your code, your work will be supervised and your instructor will take questions to make sure you understand the concepts. Programming will be done on mostly MS Windows OS, but we might explore other OS as well. Your answers to questions on the lab handouts will provide feedback and help access your work. You will be asked to formulate your own conclusions in projects and lab worksheets and we shall use discussions, presentations or lectures to present/debate these conclusions. The lab assignments, projects, worksheets, and lab final represent 36% of your total grade.

Lab grading:

Programming projects -- 180 points
Graded lab worksheets -- 120 points
Mandatory lab final -- 60 points

Total lab -- 360 points

Cheating/Plagiarism

Cheating and plagiarism will not be tolerated in any shape or form in this class. Unless you are prepared to do your own work, DO NOT take this class. Zero will be given to an assignment for the first offense and a second offense would result in a grade of "F" for the course. In addition, you may be subject to Mt. San Antonio College's student discipline process. This policy is applied to both the copier and/or the provider so protect your work. You cannot work together on any assignments unless directed by your instructor. Academic dishonesty includes but is not limited to cheating on a test, sharing a solution, sharing code, and using unauthorized code. It is not okay to let someone look at your code! When in doubt, ask your instructor. In addition, refer to the school catalog for additional information.

Useful Information

Last day to drop without a "W" is 09/08/2019 and last day to drop without getting A-F grade from this class is 11/01/2019. It is your responsibility to drop from the class or you will receive a grade. Since programming is part of the course, be prepared to be very patient and willing to spend a great deal of time with and without a computer! Any C++ or Java compiler can be used to complete your assignments.

I usually check email once a day during the week so it would be best to see me during my office hours for help. For help with a program, bring a hardcopy as well as an electronic copy of your program.

We will have seating assignment so do come early in the second meeting to select your seat. Do not eat or drink in the classroom. Please advise your instructor of any special requirements within the first week. The instructor reserves the right to revise this syllabus.

Tentative Schedule

This is just a tentative schedule and is subject to change. With an exception of a few sections, we will pretty much cover chapters 1 through 8, and 10. Lectures and reading assignments will cover the materials from the required text and it is your responsibility to read the assigned materials before coming to class. Many students tell me they have a better understanding of the materials when they read those topics before the lecture.

WEEK	DATE (M)	LECTURE ACTIVITIES	LAB ACTIVITES (lecture and demonstrate code)	TEXTBOOK
1	08/26	Introduction, C++/Java Primer	Creating a C++ or Java Program	Chapters 1 & 2
2	09/02	Holiday (M), OOP & Arrays	Working with Classes and Inheritance	Chapters 2 & 3
3	09/09	Linked Lists and Recursion	Linked Lists and Recursive Algorithms	Chapters 3 & 4
4	09/16	Analysis Tools	Project 1, Analysis of Algorithms	Chapter 4
5	09/23	Stacks, Homework 1, Review	Implementing and Using a Stack	Chapter 5
6	09/30	Queues and Deques, Exam 1	Implementing and Using a Queue	Chapter 5
7	10/07	Vectors	Implementing and Using a Deque	Chapters 5 & 6
8	10/14	Lists and Sequences	Project 2, Implementing and Using a List or Sequence	Chapter 6
9	10/21	General Trees and Traversals	Trees and Traversal Algorithms	Chapter 7
10	10/28	Binary Trees, Homework 2	Implementing a Linked Binary Tree	Chapters 7
11	11/04	Binary Search Trees, Review, Exam 2	Implementing a Binary Search Tree	Chapter 10
12	11/11	Holiday (M), AVL Search Trees	Project 3, Implementing an AVL Search Tree	Chapter 10
13	11/18	Other Search Trees – Splay Trees and (2-4) Trees	Working with Advanced Search Trees	Chapter 10
14	11/25	Priority Queues, Homework 3	Working with List or Sorted PQ	Chapter 8
15	12/02	Heaps, Review	Project 4, Working with Heap PQ	Chapter 8
16	12/09	Written Final – Wednesday @ 1:30 pm	Lab Final -- Monday @ 1:30 pm	

Last updated on 08/01/2019 10:13:37